Задания к курсу
“Разработка прикладных компьютерных систем”
3 курс, 2020-2021 уч. г. (Часть 2)

5. Добавьте в библиотеку, реализованную в задании 3 из части 1, следующие классы:
   - BaseViewModel: класс базовой модели отображения, реализующий интерфейс INotifyPropertyChanged и предоставляющий защищённые методы

     OnPropertyChanged(string propertyName)

     и

     OnPropertiesChanged(params string[] propertiesNames);
   - ConverterBase: базовый класс конвертера значений, пронаследованный от класса MarkupExtension и реализующий интерфейс IValueConverter в формате абстрактного метода Convert и виртуального метода ConvertBack, по умолчанию генерирующего исключение NotImplementedException;
   - MultiConverterBase: базовый класс конвертера значений, пронаследованный от класса MarkupExtension и реализующий интерфейс IMultiValueConverter в формате абстрактного метода Convert и виртуального метода ConvertBack, по умолчанию генерирующего исключение NotImplementedException;
   - Набор стандартных конвертеров значений: BoolToVisibility, NullToBool, NullToVisibility, Percentage, MultiBoolToBool, MultiBoolToVisibility, пронаследованных от ConverterBase и MultiConverterBase;
   - RelayCommand: класс команды, реализующий интерфейс ICommand, конструктор которого принимает на вход два делегата: действие команды и проверку на возможность выполнения действия;
   - NavigationManager: класс, хранящий ссылку на NavigationManager из объекта типа Frame и декорирующий методы перехода между содержимым Frame при помощи методов

     bool Navigate<TView>(BaseViewModel),
     bool Navigate<TView, TViewModel>(),
     bool GoBack()

     и свойства

     bool CanGoBack { get; },

     где TView - тип, производный от Page с наличием конструктора без параметров, TViewModel - тип, производный от BaseViewModel с наличием конструктора без параметров; возвращаемое значение для методов - успешно ли выполнение операции перехода.

Продемонстрируйте работу менеджера навигации на примере нескольких страниц с элементами управления, позволяющими при взаимодействии запросить переход на другую страницу / возврат по стеку страниц при помощи команд, привязанных к элементам управления..

6. Реализуйте хранение файла с видеопотоком в следующем формате:

<ширина кадра><высота кадра><частота смены кадра><код цветовой схемы><количество переходов между кадрами>[<переход между кадрами>],

где *<частота смены кадра>* - кол-во кадров, отображаемых одинаковое количество времени в рамках одной секунды видеопотока; *<код цветовой схемы>* - один из кодов:

- 0 - цветовая схема RGB;
- 1 - цветовая схема CMYK;
- 2 - цветовая схема HSB;

<переход между кадрами> - правило перехода от текущего кадра к следующему кадру, заданное в формате

<количество обновлений пикселов>[<обновление пиксела>],

где *<обновление пиксела>* - правило замены пиксела текущего кадра на пиксел следующего кадра в формате

<координата X><координата Y><компоненты цветовой схемы>,

где *X*, *Y* - координаты пиксела относительно левого верхнего пиксела изображения (имеющего координаты $X = Y = 0$, ось координат *X* направлена слева направо, ось координат *Y* - сверху вниз); *<компоненты цветовой схемы>* - компоненты цветовой схемы, заданной в заголовке файла.

Хранение файла возможно в трёх форматах: XML, JSON, двоичном. Реализуйте библиотеку с классами VideoStreamPixelRefreshRule (правило обновления пиксела), VideoStreamFrameRefreshRule (правило перехода между кадрами), VideoStream (видеопоток). Созданные классы должны поддерживать сериализацию в вышеописанные форматы хранения (а также десериализацию из них).

На основе классов для реализованного формата хранения видеопотока реализуйте оконное приложение (с применением архитектурного паттерна MVVM на основе классов из задания 5), позволяющее:

- загружать видеопоток из файла (при ошибке загрузки должна генерироваться и перехватываться исключительная ситуация собственного типа), посредством отдельного объектного сервиса (класса); выбор файла организовать через OpenFileDialog; при успешной загрузке видеопотока автоматически должен быть произведён переход к первому кадру;
- сохранять видеопоток в файл с возможностью указания формата сохранения (также посредством отдельного объектного сервиса);
- запускать и останавливать видеопоток (при завершении видеопотока он должен быть остановлен автоматически);
- отображать общую длительность видеопотока (с точностью до секунд с округлением в меньшую сторону в формате

<часы>:<минуты>:<секунды>) и время отображения текущего кадра (с отсчётом от первого кадра со временем 00:00:00);
- при помощи элемента управления Slider модифицировать скорость воспроизведения (значение коэффициента от 0.25 до 5.0 с шагом 0.25, по умолчанию 1.0);
- при помощи элемента управления Slider переходить между кадрами видеопотока.

Операции работы с файлами (сохранение/загрузка) должны быть асинхронными и поддерживать возможность отмены; во время загрузки необходимо отобразить поверх окна диалог загрузки из задания 3 (привязка свойства Visibility диалога к bool из модели отображения окна должна производиться посредством конвертера из задания 5).

7. Добавьте в задание 6 следующие возможности:
    - разложение видеопотока на несколько видеопотоков, каждый из которых сохраняет значение биективной ему цветовой компоненты, инициализируя остальные компоненты по умолчанию, с возможностью асинхронного и многопоточного построения и сохранения полученных видеопотоков в файлы (с возможностью отмены операции);
    - объединение нескольких видеопотоков в один видеопоток посредством выполнения операции сложения по модулю 2 между стоящими на одной и той же позиции относительно начала видеопотока кадрами; при различном количестве кадров во входных видеопотоках, выходной видеопоток должен иметь количество кадров равное максимальному из количеств кадров входных видеопотоков; недостающие кадры входных видеопотоков неявно должны быть заполнены чёрным цветом; частота выходного видеопотока есть частота, указанная в первом переданном файле с видеопотоком; цветовая схема выходного видеопотока также есть цветовая схема из первого переданного файла; операция объединения должна выполняться асинхронно и поддерживать отмену по запросу пользователя;
    - применения к видеопотоку для асинхронного построения видеопотока и его сохранения в файл (с возможностью отмены операции) фильтров (параметры фильтров могут быть настроены пользователем приложения):
        - выделения границ: Собела, Щарра, Превитта, *Кэнни (optional)*;
        - повышения локального контраста (лапласиан гауссиана);
        - размытия (нижних частот);
        - изменения яркости;
        - насыщенности;
    - определения наличия “движения” в видеопотоке при помощи motion detection, с возможностью указания порогового значения “разности” соседних кадров (от 0.01 до 1.00); для текущего кадра, при наличии “движения” относительно предыдущего кадра, элемент управления, имитирующий лампочку, должен загореться зелёным цветом, при отсутствии “движения” - красным цветом (реализовать через триггер

данных); motion detector можно включить/отключить в произвольный момент времени работы приложения (лампочка должна гореть синим цветом при отключённом motion detector);
- конвертацию видеопотока в видеопоток со сменой цветовой схемы (между RGB, CMYK, HSB) (асинхронно с возможностью отмены операции);
- конвертацию файла видеопотока в файл со сменой формата хранения (между XML, JSON, binary) (асинхронно с возможностью отмены операции);
- работу с несколькими видеопотоками единовременно в рамках одного экземпляра приложения при помощи контейнера TabControl, с возможностью создания и закрытия вкладок через переопределённые Header'ы TabItem'ов: при нажатии на заголовок последней вкладки должна создаться новая вкладка; при нажатии на кнопку с крестиком на любой вкладке, кроме последней (примените шаблон для Button с изображением из задания 3), вкладка должна быть закрыта. При запуске операции запуска или преобразования видеопотока на одной вкладке, должен быть возможен переход между вкладками и выполнение операций в рамках других вкладок.

Запуск выполнения операций (кроме motion detection) допускается только при остановленном видеопотоке. При выполнении операций (кроме motion detection) UI отдельной вкладки должен блокироваться диалогом загрузки с сообщением "Выполнение операции "<название операции>"", при нажатии на затемнённую область которого должна быть произведена отмена асинхронно запущенной операции.

8. Реализуйте оконное приложение, позволяющее асинхронно и многопоточно шифровать входные файлы последовательно несколькими алгоритмами (DES, Triple DES, Rijndael, RSA). Приложение должно позволять добавить в произвольное место цепочки вызов операции шифрования одним из вышеперечисленных алгоритмов, удалять вызовы из цепочки, указывать ключи шифрования, запускать процесс шифрования и дешифрования (с возможностью отмены). Запрещено использовать собственные реализации алгоритмов, необходимо использовать типы из пространства имён System.Security.Cryptography.